

SDEV1201

Database Fundamentals

LESSON 13

Let's review

What Are Data Types?

Unlike humans, a computer does not know the difference between "1234" and "abcd." A data type is a classification that dictates what a variable or object can hold in computer programming. Data types are an important factor in virtually all computer programming languages. Each data type requires different amounts of memory and has specific operations that can be performed over it.

CREATE TABLE **minimum syntax**

11

```
CREATE TABLE TableName (  
    Column1 DATATYPE  
    , Column2 DATATYPE  
    , Column3 DATATYPE  
    ...  
)
```

where *Column#* represents the name of the attribute/column and *DATATYPE* is *INT*, *DATETIME*, etc.

Constraints are used to:

21

1. Define the **PK**
 - This is defined by the **primary key** constraint
2. Define **relationships**
 - This is defined by the **foreign key** constraint
3. Define **default values**
 - Defined by the **default** constraint
4. Define a **domain of valid values**
 - Defined by the **check** constraint
5. Ensure that **all values in a column are unique**

Primary Key Constraint

When using a normalized database design, all tables must have a primary key constraint. Any column that acts as a primary key must be defined as a not null column. The DBMS will ensure that all rows in the table have a unique value for primary key columns.

When a primary key constraint is defined, PostgreSQL automatically creates a unique index for the column(s) acting as the primary key.

Syntax

Column constraint syntax:

[Constraint constraint_name] Primary Key

Table constraint syntax:

[Constraint constraint_name]

Primary Key (col_name [, col_name2 [, . . . col_name32]])

FK Constraints & referential integrity

The foreign key constraint defines referential integrity between two tables. Operations that affect the tables and/or the data stored within the tables must comply with referential integrity. This affects:

1. Dropping tables.
2. Creating tables.
3. Inserting/Updating/Deleting rows in tables.

The bottom line for referential integrity is that a child row/table cannot exist without its associated parent row/table. Therefore:

1. You must Drop a child table BEFORE you Drop the associated parent table.
2. You must Create a parent table BEFORE you Create the associated child table.
3. The value of a column acting as a foreign key must be:
 - A value that exists as a primary key in the associated parent table, or
 - NULL

Foreign Key Constraint Syntax

Column constraint syntax:

[Constraint constraint_name]
References ref_table (ref_column)

Table constraint syntax:

[Constraint constraint_name]
Foreign Key (column_name [, . . . column_name32])
References ref_table (ref_column [, . . . ref_column32])

- column_name identifies the column(s) acting as the foreign key
- ref_table identifies the parent table
- ref_column identifies the primary key in the associated parent table

Indexes

An index is an object stored in the db.

It's associated with 1 or more columns, in a specific table, that act as **the key for the index**.

This helps the DBMS look up (retrieve) the data quicker.

Functionally an index object is similar to an index at the back of a book.

Syntax

```
Create [Unique] Index [If Not Exists] index_name  
    On table_name ( column_name [, ...n] [ Asc | Desc] );
```

Column name represents the key for the index. One or more columns may act as the key for the index. All indexes must have a unique name. In SDEV1201 we normally use the prefix **IX** when naming an index.

Default Constraint

The default constraint lets you define a value that is assigned to a column when the user does not specifically supply a value when a new row is added to the table (using the Insert statement). A default constraint can be defined on any column except a column with a serial property.

The value of the default can be supplied via a constant (5.90) or a function (i.e. Current_Date). The default constraint name should use a prefix of DF (indicating this is a default constraint).

Syntax:

[Constraint constraint_name]
Default {constant | function}

Default Constraint

Examples:

Default	Constraint Definition
Use the current date as the default for the DateReceived column	Constraint DF_DateReceived Default Current_Date
Use 5.90 as the default for the HourlyRate column	Constraint DF_HourlyRate Default 5.90

Unique Constraint

The unique constraint makes sure that the values in one or more columns are not repeated. When applied to a single column, this ensures all entries in that column are different. When applied to multiple columns, this requires that the combination of values across those columns is unique, so no two rows can have exactly the same values.

A unique constraint does not consider null values to be equal.

Unique Constraint

Adding a unique constraint will automatically create a unique index on the column or group of columns used in the constraint.

Column Constraint Syntax

[Constraint constraint_name] Unique

Table Constraint Syntax

[Constraint constraint_name]

Unique (col_name [, col_name2 [, . . . col_name32]])

Check Constraint

The **CHECK** constraint enables you to specify what values are acceptable (i.e. define a DOMAIN).

CHECK constraints should be given a meaningful name and consist of an expression that evaluates to TRUE or FALSE.

Syntax:

```
CONSTRAINT CK_ConstraintName CHECK (expression)
```

Whenever a new row of data is added to the table, or an existing column has its value changed the new data will be evaluated using the expression in the check constraint. If the expression evaluates to true the DBMS will add the new data to the table. If the expression evaluates to false an error message is issued and the new data is rejected.

Check Constraint Examples

Column	Domain	Check Constraint
QuantitySold	Must be positive	Constraint CK_QuantitySold Check (QuantitySold > 0)
DateReceived	Must be >= the Date Ordered	Constraint CK_DateReceived Check (DateReceived >= DateOrdered)
CourseMark	Must be >= 0 and <= 100	Constraint CK_CourseMark Check (CourseMark Between 0 and 100) Or Constraint CK_CourseMark Check (CourseMark >= 0 and CourseMark <= 100)
PostalCode	Must follow the pattern A9A 9A9	Constraint CK_PostalCode Check (PostalCode ~ '[A-Z][0-9][A-Z] [0-9][A-Z][0-9]')

ALTER TABLE

- The **ALTER TABLE** statement is used to:
- Add a column (with or without constraints)
- Drop an existing column (assuming you do not break any referential integrity such as PK - > FK)
- Change the datatype of an existing column
- Add a default value to an existing column
- Drop a default from an existing column
- Add/Drop a Not Null constraint to an existing column
- Drop an existing constraint
- Add a table constraint
- Rename a column, or column constraint, or table

Alter Table table_name

{

Add [Column] column_name data_type [column_constraint [...]]

|

Drop [Column] column_name

|

Alter [Column] column_name Set Data Type data_type

|

Alter [Column] column_name Set Default expression

|

Alter [Column] column_name Drop Default

|

Alter [Column] column_name { Set | Drop } Not Null

|

Drop Constraint constraint_name

|

Add table_constraint

} [, ...]

Syntax

Question 3 Solution for “01 Create Table PostgreSQL

Create Table Vehicles

```
(  
  VehicleNumber integer Not Null Constraint PK_Vehicles_VehicleNumber Primary  
                                     Constraint NN_Vehicles_VehicleNumber Not Null  
  DriverFirstName varchar (35) Constraint NN_Vehicles_DriverFirstName Not Null  
  DriverLastName varchar (35) Constraint NN_Vehicles_DriverLastName Not Null  
);
```

Create Table Deliveries

```
(  
  DeliveryNumber integer Not Null Constraint PK_Deliveries_DeliveryNumber Primary Key  
                                     Constraint NN_Deliveries_DeliveryNumber Not Null,  
  DeliveryDate date Constraint NN_Deliveries_DeliveryDate Not Null,  
  VehicleNumber integer Constraint NN_Deliveries_VehicleNumber Not Null  
                                     Constraint FK_Deliveries_VehicleNumber_to_Vehicles_VehicleNumber  
                                     References Vehicles (VehicleNumber)  
);
```

3. Add the following constraints to the existing Create Table statements:

Table	Column(s)	Default	Check	Unique
Orders	OrderDate	Current Date		
Orders	PaidYN	N		
OrderDetails	Quantity		>= 0	
Customers	PostalCode		Z9Z9Z9	
Customers	PhoneNumber			Yes

Z means any upper-case character from A to Z, 9 means any digit from 0 to 9

Question 3 Solution for “01 Create Table PostgreSQL

Create Table Customers

```
(
  CustomerNumber integer Constraint NN_Customers_CustomerNumber Not Null
                        Constraint PK_Customers_CustomerNumber Primary Key,
  FirstName varchar (35) Constraint NN_Customers_FirstName Not Null,
  LastName varchar (35) Constraint NN_Customers_LastName Not Null,
  StreetAddress varchar (35) Constraint NN_Customers_StreetAddress Not Null,
  City varchar (35) Constraint NN_Customers_City Not Null,
  Province char (2) Constraint NN_Customers_Province Not Null,
  PostalCode char (6) Not Null Constraint CK_Customers_PostalCode Check (PostalCode ~ '[A-Z][0-9][A-Z][0-9][A-Z] [0-9]'),
  PhoneNumber char (12) Constraint NN_Customers_PhoneNumber Not Null
                        Constraint UQ_Customers_PhoneNumber Unique
);
```

Create Table Products

```
(
  ProductNumber char (7) Constraint NN_Products_ProductNumber Not Null
                        Constraint PK_Products_ProductNumber Primary Key,
  Description varchar (50) Constraint NN_Products_Description Not Null,
  CurrentPrice decimal (7,2) Constraint NN_Products_CurrentPrice Not Null
);
```

3. Add the following constraints to the existing Create Table statements:

Table	Column(s)	Default	Check	Unique
Orders	OrderDate	Current Date		
Orders	PaidYN	N		
OrderDetails	Quantity		>= 0	
Customers	PostalCode		Z9Z9Z9	
Customers	PhoneNumber			Yes

Z means any upper-case character from A to Z, 9 means any digit from 0 to 9

Question 3 Solution for “01 Create Table PostgreSQL

Create Table Orders

```
(  
  OrderNumber integer Constraint NN_Orders_OrderNumber Not Null  
                                Constraint PK_Orders_OrderNumber Primary Key,  
  OrderDate date Constraint NN_Orders_OrderDate Not Null  
                                Constraint DF_Orders_OrderDate_Current_Date Default Current _Z means any upper-case character from A to Z, 9 means any digit from 0 to 9  
  FloristNumber integer Constraint NN_Orders_FloristNumber Not Null,  
  CustomerNumber integer Constraint NN_Orders_CustomerNumber Not Null  
                                Constraint FK_Orders_CustomerNumber_To_Customers_CustomerNumber  
                                References Customers (CustomerNumber),  
  GSTAmount decimal (7,2) Constraint NN_Orders_GSTAmount Not Null,  
  DeliveryNumber integer Constraint NN_Orders_DeliveryNumber Not Null  
                                Constraint FK_Orders_DeliveryNumber_To_Deliveries_DeliveryNumber  
                                References Deliveries (DeliveryNumber),  
  DeliveredYN char (1) Constraint NN_Orders_DeliveredYN Not Null,  
  PaidYN char (1) Constraint NN_Orders_PaidYN Not Null  
                                Constraint DF_Orders_PaidYN_N Default 'N'  
);
```

3. Add the following constraints to the existing Create Table statements:

Table	Column(s)	Default	Check	Unique
Orders	OrderDate	Current Date		
Orders	PaidYN	N		
OrderDetails	Quantity		>= 0	
Customers	PostalCode		Z9Z9Z9	
Customers	PhoneNumber			Yes

Question 3 Solution for “01 Create Table PostgreSQL

Create Table OrderDetails

```
(  
  OrderNumber integer Constraint NN_OrderDetails_OrderNumber Not Null  
    Constraint FK_OrderDetails_OrderNumber_To_Orders_OrderNumber  
    References Orders (OrderNumber),  
  ProductNumber char (7) Constraint NN_OrderDetails_ProductNumber Not Null  
    Constraint FK_OrderDetails_ProductNumber_To_Products_ProductNumber  
    References Products (ProductNumber),  
  Quantity integer Constraint NN_OrderDetails_Quantity Not Null  
    Constraint CK_OrderDetails_Quantity_GR_EQ_Zero Check (Quantity >= 0),  
  SellingPrice decimal (7,2) Constraint NN_OrderDetails_SellingPrice Not Null,  
  Constraint PK_OrderDetails_OrderNumber_ProductNumber  
    Primary Key (OrderNumber, ProductNumber)  
);
```

3. Add the following constraints to the existing Create Table statements:

Table	Column(s)	Default	Check	Unique
Orders	OrderDate	Current Date		
Orders	PaidYN	N		
OrderDetails	Quantity		>= 0	
Customers	PostalCode		Z9Z9Z9	
Customers	PhoneNumber			Yes

Z means any upper-case character from A to Z, 9 means any digit from 0 to 9

Question 3 Solution for “01 Create Table PostgreSQL

-- Alter Table

- Add a new column called PrepaidCollect to the Deliveries table.
- The column will be one character, will be optional when inserting data.
- The only valid values will be 'P' (Prepaid) or 'C' Collect.

Alter Table Deliveries

Add PrepaidCollect Char (1) Null

Constraint CK_PrepaidCollect_PC

Check (PrepaidCollect In ('P','C'));

Question 3 Solution for “01 Create Table PostgreSQL

-- Add a new column called PaidDate to the Orders table that stores an optional date value, which should default to the current date.

Alter Table Orders

 Add PaidDate date Null

 Constraint DF_Orders_PaidDate_CurrentDate Default Current_Date;

Question 3 Solution for “01 Create Table PostgreSQL

-- Ensure that the GSTAmount in the Orders table is always greater than or equal to 0.

Alter Table Orders

Add Constraint CK_Orders_GSTAmount_GrEqZero
Check (GSTAmount >= 0);

Today's Objective

By the end of this section you will be able to:

- ☐ Write **INSERT** statements
- ☐ Write **UPDATE** statements
- ☐ Write **DELETE** statements

Reminder: Quiz 3 (Create Table & DML Statements) will be next **Monday, October 27, 2025.**

Assignment 3 (TH : Create Table & DML Statements) is available on Brightspace and is **due Friday, October 31, 2025 at 5:00 PM.**

INSERT

Insert

The Insert statement is used to add one or more rows to a table. They can contain hard coded values or can use subqueries to retrieve data from other tables to insert.

Simplified Syntax:

```
Insert [into] table_name  
      [(column list)]  
{Values  
  ({DEFAULT | NULL | expression}, . . . )  
|  
Select};
```

The INSERT Statement

The INSERT statement adds 1+ rows to a table.

```
INSERT [INTO] TableName (column1, column2, ...)  
VALUES (value1, value2, ... )
```

Values can be hardcoded values, expressions, DEFAULT, NULL, or even a subquery.

Alternatively, we can INSERT data from another table: we use the results of a SELECT query as the values to INSERT.

```
INSERT [INTO] TableName (column1, column2, ...)  
SELECT ...
```

The **INSERT** Statement

If an Insert statement supplies data that does not comply with the constraints defined for a column or supplies data that is incompatible with the data type of the column the Insert fails and an error message is issued by SQL Server.

In general, there are three ways to do an insert:

1. Select data from another table (e.g. a backup table).
2. Hard code the data values in the INSERT statement itself.
3. A combination of methods 1 and 2.

Example

Insert Into Staff

(StaffID, FirstName, LastName, DateHired, DateReleased, PositionID, LoginID)

Values

(25, 'Alex', 'Code', '2024-10-01', Null, 4, 'acode');

Note:

- The list of values is in the same order as the column list
- The keyword null is used to insert a value of null in DateReleased column.

Example

The same record could also be inserted as follows:

Insert Into Staff

Values

(30, 'Alex', 'Code', '2024-10-01', Null, 4, 'acode');

Note that:

1. The values are supplied in the order that they appear in the table definition.

****This is not the recommended method and will not be considered acceptable in this course****

Insert

Sometimes data to be inserted is selected from the database. Data to be inserted can be provided by the user, selected from the database, or a combination of them both.

Insert Into Staff

```
(StaffID, FirstName, LastName, DateHired, DateReleased,  
PositionID,  
LoginID)
```

Values

```
(35, 'Alex', 'Code', '2024-10-01', Null,  
(Select PositionID From Staff  
Where FirstName = 'Robert' And LastName = 'Smith'),  
'acode');
```

This Insert statement would insert a new record for Alex Code and assign him the same PositionID as Robert Smith.

Insert

It is important to remember that the subquery must only return 1 value to be used in the insert statement. If the subquery returns more than one value, an error will occur.

ERROR: more than one row returned by a subquery used as an expression

Insert

You can also insert a record with one insert statement with multiple subqueries (with or without delimiters) used as values:

Insert Into Staff

(StaffID, FirstName, LastName,
DateHired, DateReleased,
PositionID, LoginID)

Values

```
(  
(Select Max(StaffID) + 1 From Staff), 'Sarah', 'Lin',  
'2025-10-01', Null,  
(Select PositionID From Position  
Where PositionDescription = 'Instructor'), 'slin'  
);
```

Insert

If a column is defined with a default constraint the default can be used in the insert with the keyword default OR the column name can be omitted from the column list and no value provided for the column. Note: there are currently no defaults in the IQSchool database tables.

Insert Into Staff

```
(StaffID, FirstName, LastName, DateHired, DateReleased, PositionID, LoginID)
```

Values

```
(50, 'Alex', 'Code', Default, Null, 4, 'acode');
```

OR

Note: The DateHired column is not in the following column list:

Insert Into Staff

```
(StaffID, FirstName, LastName, DateReleased, PositionID, LoginID)
```

Values

```
(52, 'Alex', 'Code', Null, 4, 'acode');
```

Insert

You can also insert multiple records with one insert statement with all hardcoded values:

Insert Into Staff

(StaffID, FirstName, LastName, DateHired, DateReleased, PositionID,
LoginID)

Values

(55, 'Maria', 'Lopez', '2025-09-01', Null, 1, 'mlopez'),
(60, 'David', 'Kim', '2025-09-01', Null, 2, 'dkim'),
(65, 'Helen', 'Zhou', '2025-09-15', Null, 3, 'hzhou');

Activity

Please work on “Insert Exercises” found in the Activities section of “Week 7 – DML statements”. The questions are based on the IQSchool database.

Insert Exercises Solution

Question 1

-- 1. Insert a new record into the Payment table for a student named Peter Pan who paid \$450.00 using Visa on October 10, 2005. Use subqueries to look up the
PaymentTypeID and StudentID.

Insert Into Payment

(PaymentID, PaymentDate, Amount,
PaymentTypeID,
StudentID)

Values

```
(  
  40, '2005-10-10', 450.00,  
  (Select PaymentTypeID  
   From PaymentType  
   WherePaymentTypeDescription = 'VISA'),  
  (Select StudentID  
   From Student  
   WhereFirstName = 'Peter' And  
   LastName = 'Pan')  
);
```

Insert Exercises Solution

Question 2

-- 2. Add a new Registration record for student Robert Smith, who is taking course DMIT1514 in semester 2005J, with a mark of 87, WithdrawYN = 'N', from instructor Tess Agonor. Use subqueries to retrieve both the StudentID and the StaffID.

Insert Into Registration

```
(StudentID,  
 CourseID, Semester, Mark, WithdrawYN,  
 StaffID)
```

Values

```
(  
(Select StudentID  
 From Student  
 WhereFirstName = 'Robert' And  
 LastName = 'Smith'),  
'DMIT1514', '2005J', 87, 'N',  
(Select StaffID  
 From Staff  
 WhereFirstName = 'Tess' And  
 LastName = 'Agonor')  
);
```


Insert Exercises Solution

Question 3

-- 3. Insert three new clubs into the Club table in a single Insert statement using the following values:

- a. AI Artificial Intelligence Society
- b. ROBO Robotics Club
- c. CYBER Cybersecurity League

Insert Into Club

(ClubID, ClubName)

Values

('AI', 'Artificial Intelligence Society'),

('ROBO', 'Robotics Club'),

('CYBER', 'Cybersecurity League');

Insert Exercises Solution

Question 4

-- 4. Add a new student (201052481) named Helen Grant who lives at 12 Maple Ave, Spruce Grove, postal code T7X3P4, Gender F, BalanceOwing \$1,800.00 and was born on September 1, 2000.

Insert Into Student

(StudentID, FirstName, LastName, Gender, StreetAddress, City,
PostalCode, BalanceOwing, BirthDate)

Values

(201052481, 'Helen', 'Grant', 'F', '12 Maple Ave', 'Spruce Grove', 'T7X3P4',
1800.00, '2000-09-01');

Insert Exercises Solution

Question 5

-- 5. A new club called AI Enthusiasts has been created. Insert records into Activity for all students from the city of Edmonton, assigning them to the club 'AI'. Use an Insert Into ... Select statement.

```
Insert Into Activity
    (StudentID, ClubID)
Select StudentID, 'AI'
From   Student
Where  City = 'Edmonton';
```

Insert Exercises Solution

Question 6

-- 6. Insert a new payment (number 50) for Robert Smith, equal to half of his current BalanceOwing dated today. Use subqueries and expressions to get the IDs and to calculate the amount dynamically.

Insert Into Payment

(PaymentID, PaymentDate,
Amount,
PaymentTypeID,
StudentID)

Values

(50, Current_Date,
(Select BalanceOwing / 2
From Student
Where FirstName = 'Robert' And
LastName = 'Smith'),
(Select PaymentTypeID
From PaymentType
Where PaymentTypeDescription = 'Cash'),
(Select StudentID
From Student
Where FirstName = 'Robert' And
LastName = 'Smith')
);

Insert Exercises Solution

Question 7

-- 7. Enroll Isaac Newton as a new student with two separate Insert statements (one for Student, one for Registration).

-- a. StudentID: 201052500

-- b. Gender: N

-- c. Address: '45 Science Rd, Edmonton'

-- d. Postal Code: 'T5A1L9'

-- e. Balance Owing: 2,500.00

-- f. Birth Date: 1995-09-10

-- g. Register them in course DMIT1514 for semester 2005J, with instructor Tess Agonor

-- Step 1: insert student

Insert Into Student

(StudentID, FirstName, LastName, Gender, StreetAddress, City, PostalCode, BalanceOwing, BirthDate)

Values

(201052500, 'Isaac', 'Newton', 'N', '45 Science Rd', 'Edmonton', 'T5A1L9', 2500.00, '1995-09-10');

-- Step 2: register them in a course

Insert Into Registration

(StudentID, CourseID, Semester, Mark, WithdrawYN,
StaffID)

Values

(201052500, 'DMIT1514', '2005J', Null, 'N',

(Select StaffID

From Staff

Where FirstName = 'Tess' And

LastName = 'Agonor')

);

UPDATE

Update

The Update statement is used to update existing data in a table. It can be used to update one or more columns. Data can be updated by providing new values for columns or by using subqueries to provide the new data.

Update

Simplified Syntax:

```
UPDATE table_name
SET    column = expression [ , column = expression . . . ]
[FROM table_name1 [, table_name2 ...] ]
[WHERE expression];
```

table_name - identifies the table to be updated

SET - identifies the column(s) to be updated and the expression which provides the new value for the column. The expression can be a subquery.

FROM clause - optional

- identifies additional table(s) referenced in the where clause that lets you update rows in a table based on data in one or more other tables. If more than one table is included, tables are comma-delimited

WHERE clause - optional. Identifies what records are to be updated. if the FROM clause is utilized, table join condition(s) are coded here

Example

```
Update   Course  
Set      MaxStudents = 3  
Where    CourseID = 'DMIT1001';
```

Updates the Course table and sets the MaxStudents to 3 for CourseID DMIT1001.

Example

Update Course

Set CourseCost = CourseCost * 1.10

Where CourseID = 'COMP1017';

Updates the Course table and increases the cost of COMP1017 by 10%.

Example

```
Update   Course
Set      CourseCost = (Select CourseCost
                        From      Course
                        Where     CourseID = 'DMIT2590')
Where    CourseID = 'DMIT2515';
```

Updates the Course table and sets the cost of DMIT2515 to be the same cost as DMIT2590.

Example

More than one column can be updated in a single update statement.

```
Update    Course
Set       CourseHours = 4,
          MaxStudents = 5,
          CourseCost= 300
Where     CourseID = 'DMIT1001';
```

Updates CourseHours, MaxStudents and CourseCost columns for CourseID DMIT1001.

Activity

Please work on “Update Exercises” found in the Activities section of “Week 7 – DML statements”. The questions are based on the IQSchool database.

Reminder

Quiz 3 (Create Table & DML Statements) will be next **Monday, October 27, 2025**.

Assignment 3 (TH : Create Table & DML Statements) is available on Brightspace and is **due Friday, October 31, 2025 at 5:00 PM**.

Update Exercises Solution

Question 1

-- Update Exercise Solution

-- Use the IQSchool Database

-- 1. Update Staff member Robert Smith's LoginID to rsmith25.

Update Staff

Set LoginID = 'rsmith25'

Where StaffID = (Select StaffID

From Staff

Where FirstName = 'Robert' And

LastName = 'Smith');

Update Exercises Solution

Question 2

-- 2. Increase all course hours by 5.

Update Course

Set CourseHours = CourseHours + 5;

Update Exercises Solution

Question 3

-- 3. Reduce Minnie Ono's BalanceOwing by 100.

```
Update Student
Set BalanceOwing = BalanceOwing - 100
Where StudentID = (Select StudentID
                   From Student
                   Where FirstName = 'Minnie' And
                           LastName = 'Ono');
```


Update Exercises Solution

Question 4

-- 4. Change all students born before 1990 to have BalanceOwing = 0.

Update Student

Set BalanceOwing = 0

Where Birthdate < '1990-01-01';

Update Exercises Solution

Question 5

-- 5. Increase the CourseCost by 10% for all courses taught by staff with the PositionDescription of Instructor.

Update Course

Set CourseCost = CourseCost * 1.10

Where CourseID In (Select Registration.CourseID

From Registration

Inner Join Staff

On Registration.StaffID = Staff.StaffID

Inner Join Position

On Staff.PositionID = Position.PositionID

Where Position.PositionDescription = 'Instructor'

);

Update Exercises Solution

Question 6

-- 6. Set the WithdrawYN column to 'Y' for all registrations where the mark is below 50, but only for 2004 or later semesters.

```
Update    Registration
```

```
Set WithdrawYN = 'Y'
```

```
Where Mark < 50 And
```

```
    Cast(Left(Semester,4) As Integer) >= 2004;
```

-- or --

```
Update    Registration
```

```
Set WithdrawYN = 'Y'
```

```
Where Mark < 50 And
```

```
    Cast(Substring(Semester,1,4) As Integer) >= 2004;
```

Update Exercises Solution

Question 7

-- 7. Update the Student record for Peter Codd to show a new address: 100 Main Street, Edmonton, AB, T5B3G5.

Update Student

Set StreetAddress = '100 Main Street',

City = 'Edmonton',

Province = 'AB',

PostalCode = 'T5B3G5'

Where FirstName = 'Peter' And

LastName = 'Codd';

Reminder

Quiz 3 (Create Table & DML Statements) will be next **Monday, October 27, 2025.**

Assignment 3 (TH : Create Table & DML Statements) is available on Brightspace and is **due Friday, October 31, 2025 at 5:00 PM.**